УДК 004.056.5

В. В. КАРПІНЕЦЬ, О. І. КОСТЮЧЕНКО, П. В. ПАВЛОВСЬКИЙ, А. В. ПРИЙМАК, С. В. ЮХИМЕНКО

ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОСТІ КОРИСТУВАЧА ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ДО ІНФОРМАЦІЇ ЗАСОБАМИ ГІБРИДНОГО ГІПЕРВІЗОРА

Вінницький національний технічний університет,
21021, вул. Хмельницьке шосе, 95, м. Вінниця, Україна,
E-mail: karpinets@gmail.com

Анотація. В роботі запропоновано систему захисту від несанкціонованого доступу на основі ізольованої паравіртуалізації з використанням гібридного гіпервізора, низки операційних систем та сформованими правилами користування даними інструментами.

Ключові слова: Паравіртуалізація, гібридний гіпервізор, віртуалізація, несанкціонований доступ

Аннотация. В работе предложена система защиты от несанкционированного доступа на основе изолированной паравиртуализации с использованием гибридного гипервизора, ряда операционных систем и сформированными правилами пользования данными инструментами.

Ключевые слова: паравиртуализация, гибридный гипервизор, виртуализация, несанкционированный доступ

Abstract. This work proposes a system of protection against unauthorized access on the basis of isolated paravirtualization using a hybrid hypervisor, a number of operating systems and the established rules for the use of these tools.

Key words: Paravirtualization, hybrid hypervisor, virtualization, unauthorized access

ВСТУП

Оскільки практично будь-яка операційна система має помилки, запасні ходи, недопрацювання, надлишковості, вразливості, а деякі навіть мають завідомо недостатньо надійну політику безпеки, то майже неможливою задачею є розробка ефективної системи безпеки на основі однієї конкретної робочої операційної системи [1]. Разом з цим, кількість векторів атаки на користувача операційних систем лише зростає з кожним роком, а складність та продуманість конкретних методів взлому виходять на все нові і нові рівні, що ускладнює боротьбу з ними традиційними методами.

Зокрема, шкідливе програмне забезпечення може проникнути через веб-переглядач за допомогою скриптів та ін'єкцій, навіть якщо користувач системи не завантажує навмисне жодних файлів із переглядача [2]. До того ж, новостворені модифікації шкідливого програмного забезпечення потрапляють в антивірусні бази із запізненням в кілька днів, що робить антивірусні комплекси вразливими до останніх версій шкідливого програмного забезпечення і нових його видів.

Сьогодні доступні інструменти для ефективного створення, налаштування та використання ізольованих віртуальних машин на одній фізичній машині. За допомогою них можна безпечно розмежувати діяльність користувача на ізольовані домени різних рівнів безпеки. А потенційно небезпечні дії можна виконувати у одноразових доменах, на яких завідомо немає ніяких важливих даних. Ураження одного домену не впливає на безпеку інших доменів, тому такий підхід до впровадження безпеки в змозі забезпечити захищеність від великої кількості загроз, в тому числі від вразливостей нульового дня, на відміну від традиційних антивірусних комплексів.

Хоча суттєвого рівню інформаційної безпеки досягти в рамках класичного підходу практично майже неможливо, проте є можливість значно підвищити інформаційну безпеку, використовуючи не єдину операційну систему, а їх комплекс, розмежовуючи діяльність користувача в ізольованих середовищах таким чином, щоб користувач зміг виконувати дії різного рівню безпеки у відповідних різних середовищах[3].

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

В рамках класичного підходу до забезпечення безпеки користувача ЕОМ від несанкціонованого доступу до інформації, котрий є найпопулярнішим підходом на сьогодні, прийнято застосовувати антивірусне програмне забезпечення як основний інструмент захисту[5]. Даний підхід має свої недоліки в аспекті інформаційної безпеки.

На жаль, традиційні підходи до безпеки, такі як антивірусні програми та програмні або апаратні брандмауери, на сьогоднішній день, не є достатніми, щоб уникнути витончених зламів. Наприклад, нині творці зловмисного програмного забезпечення дуже часто перевіряють, чи їх шкідливе програмне забезпечення не розпізнано будь-якими антивірусними програмами на основі програмних підписів. Якщо ПЗ розпізнається, вони перетворюють код до тих пір, поки антивірусні програми більше не впізнають його, а потім поширюють змінене ПЗ. Бази антивірусних програм регулярно оновлюються, але нова версія шкідливого ПЗ потрапляє до бази зазвичай, щонайменше, через декілька днів після того як починає поширюватись. Це надто пізно по відношенню до тих комп'ютерів, які вже було скомпрометовано. Більш просунуте антивірусне програмне забезпечення може працювати краще і швидше у цьому аспекті, але все одно обмежується даним підходом до виявлення. Нові вразливості нульового дня постійно виявляються у загальнодоступному програмному забезпеченні, яке широко використовується; наприклад, у веб-переглядачах, офісному ПЗ, клієнтах файло-обмінників, та багатьох інших, а антивірусна програма або брандмауер не можуть запобігти використанню всіх цих вразливостей.[6]

Метод аналізу підозрілої поведінки програмного забезпечення теж має недоліки, які не дозволяють йому забезпечити високу безпечність системи. Досить часто не зловмисне ПЗ має ознаки зловмисного, намагаючись працювати із виконуваними файлами та бібліотеками, із реєстрами, роблячи резервні копії самого себе, скануючи файлову систему для власних потреб тощо.

Через велику кількість хибних спрацювань антивіруса у відповідь на подібні дії ПЗ, користувачу не спеціалісту важко відрізнити хибне спрацювання від дійсного, що спричиняє заборону роботи доброякісного ПЗ, або дозвіл на роботу шкідливого ПЗ. Окрім того, шкідливе ПЗ може користуватись недокументованими вразливостями операційної системи, і внаслідок цього виконувати шкідливі дії так, щоб антивірус не розцінив ці дії як шкідливі.[7]

Проте, є можливість розробити систему розподілу діяльності користувача ЕОМ по групам із різними рівнями безпеки, що дозволить ізолювати більш безпечні задачі від менш безпечних.

Це можна зробити за допомогою комп'ютерної віртуалізації і безпечно реалізувати подібний розподіл на одній ЕОМ. Враховуючи особливості різних підходів до віртуалізації, для вирішення поставленої проблеми, було обрано віртуалізацію платформи, як найбільш швидкий і гнучкий у налаштуванні і при цьому не менш безпечний, ніж альтернативні підходи. Також було з'ясовано, що серед технологій віртуалізації платформи найбільш придатною для поставленої задачі є технологія паравіртуалізації, оскільки їй характерні одночасно винятково висока швидкодія і високий рівень безпеки, притаманні апаратній віртуалізації, але при цьому паравіртуалізація ще й значно більш універсальна у застосуванні і гнучка для налаштування, подібно програмній віртуалізації. Оскільки єдиним типом гіпервізорів, який підтримує паравіртуалізацію є гібридний тип, то систему безпеки необхідно будувати саме на основі гібридного гіпервізора.

Тому актуальним є створення системи захисту користувача від несанкціонованого доступу до інформації, котра підвищить стійкість системи користувача до атак зловмисника за рахунок поєднання різних операційних систем довільної кількості ізольовано одна від одної у віртуальних машинах, та виконання задач різних сфер діяльності у різних віртуальних машинах засобами гібридного гіпервізора, що призводить до суттєвого підвищення безпеки кожної окремої віртуальної машини.

1. РОЗРОБКА СИСТЕМИ ЗАХИСТУ КОРИСТУВАЧА ВІД НСД ДО ІНФОРМАЦІЇ ЗАСОБАМИ ГІБРИДНОГО ГІПЕРВІЗОРА

Гіпервізор Xen Project [8], або просто Xen — гіпервізору типу 1+ (гібридний гіпервізор), що робить можливим запуск багатьох екземплярів операційної системи, або запуск фундаментально різних операційних систем паралельно на одній машині. Xen — також єдиний гіпервізор типу 1+ із відкритим програмним кодом. Він розроблений в комп'ютерній лабораторії Кембриджського університету і поширюється на умовах вільної ліцензії GNU GPL 2. Даний гіпервізор використовується в якості основи для ряду різних комерційних та відкритих проектів наступних типів: віртуалізація серверів, інфраструктура як служба (IaaS, англ. Infrastructure as a Service), віртуалізація на настільному комп'ютері, програми безпеки, вбудовані та апаратні пристрої.

Архітектура Xen має сильнішу ізоляцію між гіпервізором і гостьовим ядром в порівнянні з іншими гіпервізорами (рис. 1).

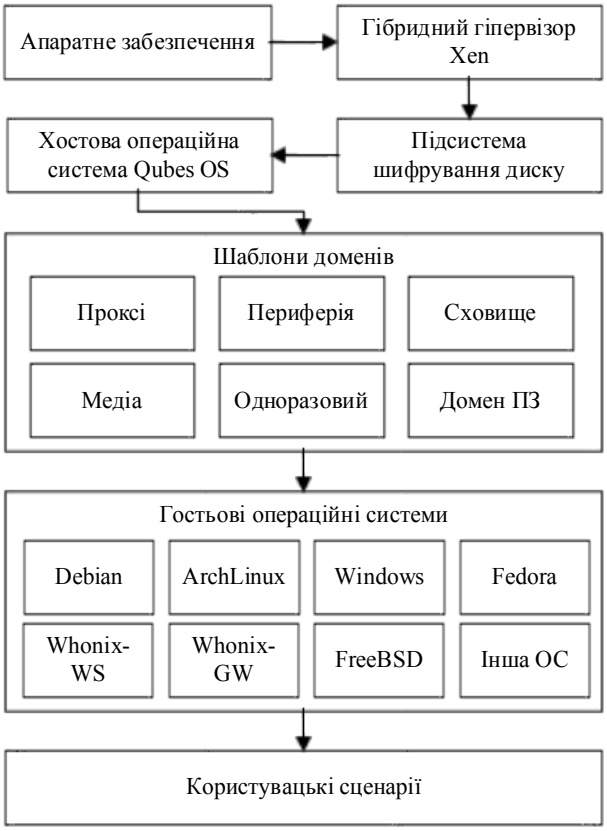

Рис. 1 Структурна схема ієрархії компонентів розробленої системи захисту

Також в ньому є наступні особливості, котрі є перевагами в аспекті безпеки, котрі більш комплексно відділяють систему для зменшення ризику зараження програмними закладками (експлойтами):

1) таблиці грантів керують доступом до пам'яті, котра виділена для кожної віртуальної машини, а також дозволяють контролювати розподілення пм'яті між віртуальними машинами, створюючи більш надійну межу безпеки довкола них і роблячи неможливим перехід експлойтів із однієї віртуальної машини на іншу;

2) дезагрегування Xen дозволяє ізолювати окремі основні функції гіпервізора в окрему віртуальну машину (майже завжди віртуальні машини використовують як процеси, в якості наступного рівня ізоляції). Особливий варіант використання — так називаний режим «Secure I/O», котрий дозволяє запускати окремі драйвери пристроїв на віртуальних машинах.

3) функція самодіагностики віртуальної машини (VMI), котра є ексклюзивною для Xen. Вона знаходить нові класи загроз безпеки системи і вже використовується в ряді різноманітних комерційних продуктів.

Самодіагностика віртуальної машини (VMI) в Xen є новим підходом до виявлення загроз безпеки, дозволяючи контролювати пам'ять віртуальних машин поза віртуальною машиною, практично виключаючи необхідність втручання. Якщо виявлена підозріла активність, така як атака нульового дня, користувацьке програмне забезпечення для моніторингу може прийняти запобіжні міри для їх усунення.

Якщо потрібно провести виявлення шкідливих програм і загроз в типовій віртуалізованій системі, то адміністратору такої системи необхідно запустити агент (наприклад, антивірус, інспектор мережевих пакетів і т.д.) в кожній окремій віртуальній машині. Перша проблема, пов'язана з цим процесом, полягає у тому, що агенту необхідно буде відсканувати пам'ять, сховище і т.д., так що цей метод є досить інтрузивним. Друга проблема полягає в тому, что якщо в операційній системі. Встановленій на віртуальній машині, є експлойт, то руткіт (rootkit) або просунута постійна загроза (APT) можуть взяти під контроль ядро і обманути агента, котрий вважатиме, що система не заражена. Тому при використанні традиційної моделі безпеки руткіт чи APT можуть вимкнути систему користувача, якщо в ОС, встановленій на його віртуальній машині, є експлойт.

За допомогою VMI можна створювати спеціальну віртуальну машину під назвою «Додаток безпеки» (Security Appliance), котра містить програмне забезпечення безпеки і працює з інтерфейсом VMI. Цей інтерфейс налаштований для моніторингу пам'яті декількох віртуальних машин. Також можливо розбити великі системи для підвищення масштабованості. Наприклад, на одному хосту можуть існувати кілька різноманітних движків самодіагностики, контролюючих різні віртуальні машини.

Дійсно корисною функцією VMI є те, що користувач може створювати правила, котрі виявляють конкретні техніки атак, такі як «виконання коду в купі». Це можна перетворити в правило у VMI, котре буде інформувати клієнта, працюючого в Security Appliance, коли код виконується в областях пам'яті, пов'язаних із кучею. Як тільки така подія буде мати місце, пристрій безпеки зможе вжити заходів щодо усунення порушень.

Оскільки різноманітні віруси і шкідливі програми використовують відносно невелику кількість технік прикріплення (наприклад, переповнення буферу, виділення додаткової пам'яті під кучу, ін'єкція коду, прив'язка API), можна створити правила для VMI, котрі би дозволяли виявити ці техніки. Це дозволить захищатись від нових класів шкідливих програм. Навіть не маючи відомостей про те, що вони роблять із системою, на відміну від традиційних інструментів перевірки наявності вірусів чи шкідливих програм, котрі покладаються на перевірку сигнатур шкідливих програм чи слідів артефактів, котрі шкідливе ПЗ залишає в системі.

Самодіагностика віртуальної машини — це метод виявлення конкретних методів атаки, котрий характеризується низьким степенем втручання.

На рис.2 показано, що VMI також працює за межами віртуальної машини, тобто є можливість уникнути раніше згаданого сценарію, в якому руткіти або APT можуть обманути конкретне програмне забезпечення безпеки. Тим не менше, VMI не заміняє традиційні рішення безпеки, тому що на даний момент вона не може захистити проти всіх класів атак, не впливаючи на швидкодію системи. В майбутньому це стане можливим, але все ж по міркам безпеки рекомендується використовувати VMI із традиційними заходами для підтримання безпеки.

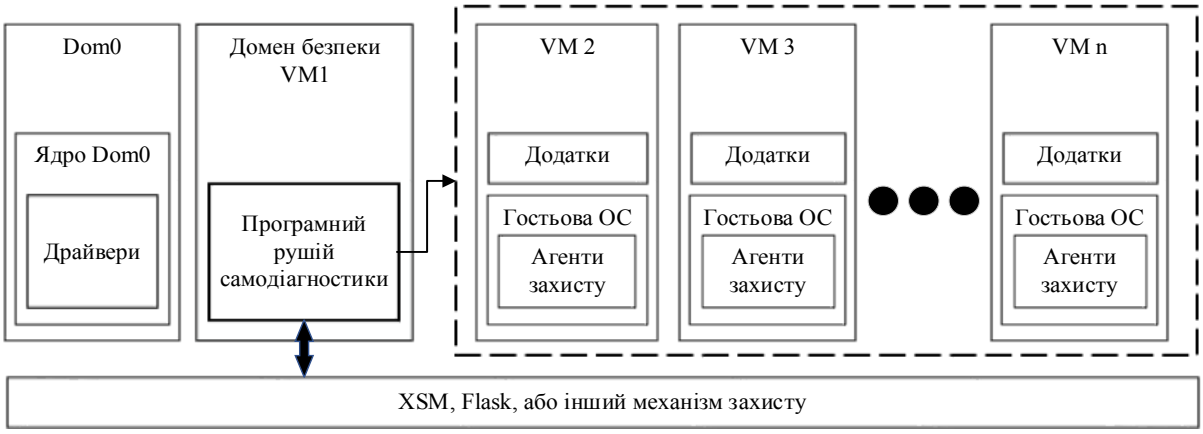

Рис. 2. Схема роботи VMI

Драйвери пристроїв в архітектурі Xen завжди запускаються в Dom0. На приведеній на рис.2 схемі дезагрегації із схеми було видалено ці драйвери із Dom0 і розміщено їх у мережевий домен (Network Driver), працюючий на звичайній віртуальній машині, не маючій привілегій. Це означає, що Dom0 був видалений із шляху передачі даних для роботи в мережі. Якщо би на мережевий драйвер була здійснена атака, це повпливало би тільки на мережевий домен. Оскільки в цьмоу домені працює тільки драйвер пристрою, експлойт повинен був би перестрибнути через додаткову границю VM, щоб спричинити яку-небудь шкоду. Та ж модель може застосовуватись для доменів сховищ. В Dom0 залишається тільки функціональність, котра керує всією інформацією про конфігурації в системі Xen (наприклад, QEMU чи домен Xenstore). Знову ж таки, можна просто запустити їх в іншому домені Xen, котрий створить те, що називається сильно дезагрегованою системою. В дезагрегованій архітектурі драйвери пристроїв видаляються із Dom0 і розміщаються в мережевий домен (рис. 3).

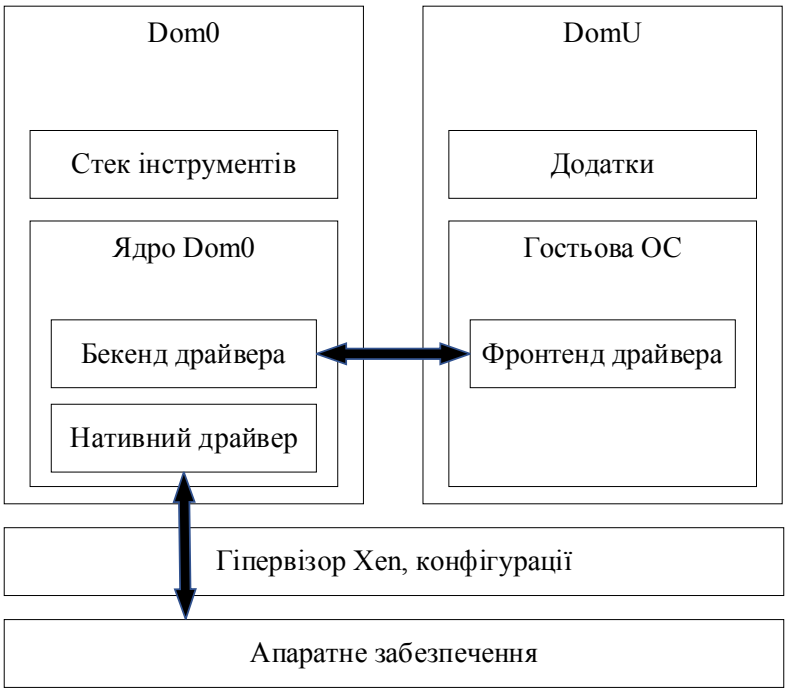

Рис. 3. Схема прикладу дезагрегованої системи і її мережі

Ця ж модель може застосовуватись для доменів сховищ. В Dom0 залишається тільки функціональність, котра керує всією інформацією про конфігурації в системі Xen (рис. 4)

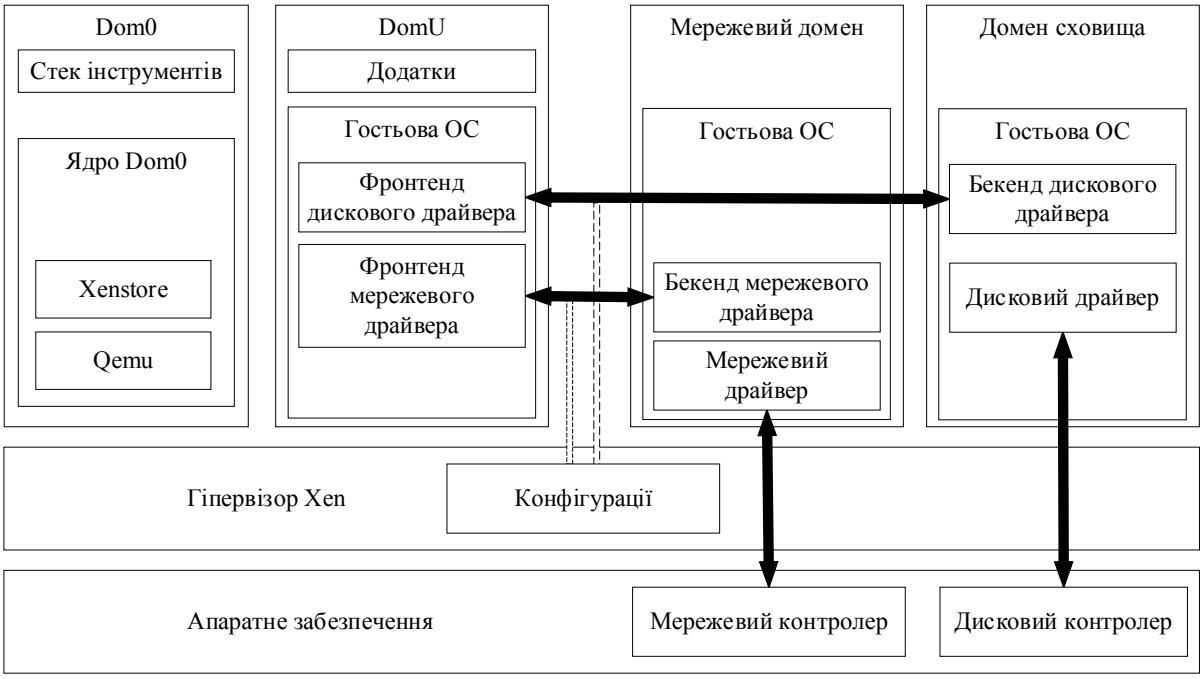

Рис. 4. Схема прикладу дезагрегованої системи і її сховища

Розглянемо детальніше Qubes OS, оскільки вона є найбільш прийнятним інструментом для вирішення задач даної роботи, тому що призначена саме для безпеко-орієнтованого користування нею користувачем локальної машини.

Qubes OS — орієнтована на безпеку операційна система для настільних комп'ютерів, котра призвана забезпечити безпеку через ізоляцію. Віртуалізація в ній втілюється на базі гіпервізора Xen. Користувацьке середовище може бути заснованим на Fedora, Debian, Whonix, Windows та інших операційних системах.

Qubes OS реалізує підхід “безпека в ізоляції”.[6] Передбачається, що не може бути ідеального безпомилкового середовища робочого столу. Таке середовище нараховує мільйони стрічок коду, мільярди програмних/апаратних взаємодій. Одна критична помилка може призвести до того, що зловмисне програмне забезпечення візьме контроль над машиною.[7]

Qubes використовує підхід, який називається безпекою шляхом розподілу, що дозволяє розділити різноманітні частини цифрової діяльності користувача на надійно ізольовані відсіки, котрі в даній операційній системі називаються “qubes” (“qube” в однині).

Цей підхід дозволяє зберігати різні дані та дії, які користувач виконує на своєму комп’ютері, надійно відокремленими один від одного в ізольованих робочих середовищах, щоб компрометація одного середовища не вплинула на інші середовища. Наприклад, у користувача може виникнути потреба в тому щоб відвідати ненадійний веб-сайт і при цьому безпечно займатись онлайн-банкінгом, тоді варто розділити ці дві дії у два ізольованих середовища. Таким чином, якщо ненадійний веб-переглядач користувача буде скомпрометовано веб-сайтом із зловмисним ПЗ, банківській діяльності користувача це не буде загрожувати. Подібним чином, якщо користувач підозрює наявність зловмисного вмісту в додатках до електронного листа, Qubes OS може відкривати кожен додаток в окремому одноразовому середовищі так, що навіть кожне одноразове середовище з додатком не зможе вплинути на інше подібне, і на інші складові системи в тому числі. Завдяки такому підходу Qubes OS дозволяє користувачеві робити будь-що на одній фізичній машині не хвилюючись за те що одна успішна кібератака вразить всю його цифрову діяльність.

Не всі програмні засоби для створення віртуальних машин рівні, коли мова йде про безпеку. Типові гіпервізори, що запускаються поверх традиційної ОС (наприклад: VirtualBox, Vmware Workstation, Hyper-V та інші), є гіпервізорами типу 2, що означає що вони за визначенням настільки ж безпечні, наскільки і їх хостова ОС. Якщо хостову ОС коли-небудь буде скомпрометовано, то фактично буде скомпрометовано будь-яку віртуальну машину гіпервізора типу 2.

На відміну від такого підходу, Qubes OS використовує гіпервізор Xen типу 1. Це означає, що зловмисник повинен бути здатен скомпрометувати сам гіпервізор, щоб поставити під загрозу всю систему, що практично є задачею значного вищого порядку складності.

2. КОРИСТУВАЦЬКІ СЦЕНАРІЇ ДЛЯ СИСТЕМИ БЕЗПЕКИ

Спроектована в даній роботі система безпеки не є монолітною та незмінною. Навпаки, вона гнучка і в ній є багато можливостей по організації довільної організації складових елементів і застосування різноманітних універсальних механізмів для досягнення конкретних цілей.

Наприклад, в рамках першого сценарію користувачеві необхідно завантажити деякі файли із мережі Інтернет, і якимось чином використати ці файли, або обробити. Але ці файли отримуються із ненадійного, або потенційно ненадійного джерела у мережі, що несе за собою ризик. Якби користувач пробував робити це у системі із традиційним підходом до безпеки, то такими діями наражав би на небезпеку одразу всю систему, оскільки у традиційних системах шкідливе ПЗ компрометує одразу всю систему цілком.

У нашому ж випадку (рис. 5), користувач має змогу завантажити потенційно небезпечні файли до одноразового середовища, або до середовища, передбаченого для роботи з файлами у подібних ситуаціях. Це створює додаткову стіну між потенційно шкідливим ПЗ і всією системою.

Користувач може заздалегідь зробити шаблон для подібного сценарію. Це може бути шаблон віртуальної машини із спеціальними налаштуваннями і спеціальним оточенням ПЗ, у тому числі із антивірусами, а може бути одноразова віртуальна машина, яка не несе у собі якоїсь особливої цінності, а створюється спеціально в якості випробувального полігону для подібних ситуацій, тому навіть якщо вона буде скомпрометована – це не грає ніякої ролі глобально, оскільки після виконання своїх цілей вона просто припинить своє існування, не залишивши слідів.

Таким чином, користувач може умовно безпечно працювати із небезпечними фалйами та ПЗ, передаючи лише результат цієї роботи у віртуальні машини із вищим рівнем безпеки, або передаючи завантажені файли у такі машини уже після того, як переконається, що це безпечно методом випробувань та перевірок у ізольованому середовищі.

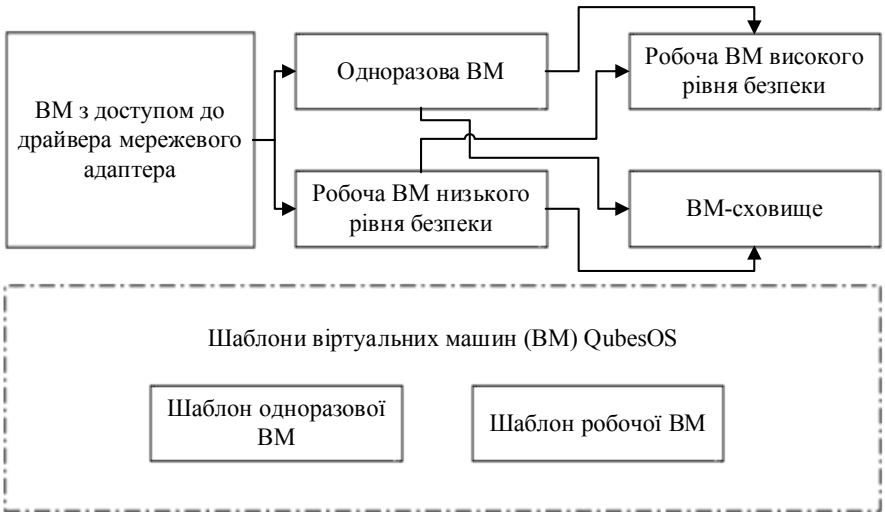

Рис.5. Схема сценарію роботи користувача системи із потенційно небезпечними файлами із мережі Інтернет

У рамках другого сценарію, нехай користувач матиме деякі конфіденційні дані, доступ до яких він бажає обмежити по тим чи іншим причинам. При цьому, час від часу йому потрібно ці дані обробляти, систематизувати, редагувати, або додавати.

В рамках традиційної системи, користувач наражає такі дані на велику небезпеку, даючи загальній системі доступ до тієї інформації, оскільки система може бути якимось чином скомпрометована.

За допомогою ж розробленої системи захисту, користувач має змогу помістити свої конфіденційні дані у спеціальний контейнер, який можна назвати сховищем. Віртуальна машина сховища не матиме доступу до локальних та глобальних мереж, у ній не будуть працювати програми із сумнівною надійністю, а отже це сховище буде відносно надійним, щоб вдало виконувати свої функції по безпечному збереженню даних.

В рамках запропонованого сценарію (рис. 6) користувач зберігатиме свої конфіденційні дані у сховищі, а уже із нього, за потреби, передаватиме ці дані у інші віртуальні машини, в яких буде здійснюватися їх обробка. Тобто, користувач матиме можливість дозовано передавати порції даних, відбираючи лише необхідні до передавання зараз, на обробку. Наприклад, такою обробкою може бути систематизування таблиць із сховища у спеціально створеній для цього машині, а потім графічна обробка отриманих даних у іншій спеціалізованій машині, після чого повернення результатів у сховище та відвантаження у мереже, якщо це потрібно, через іншу спеціалізовану віртуальну машину.

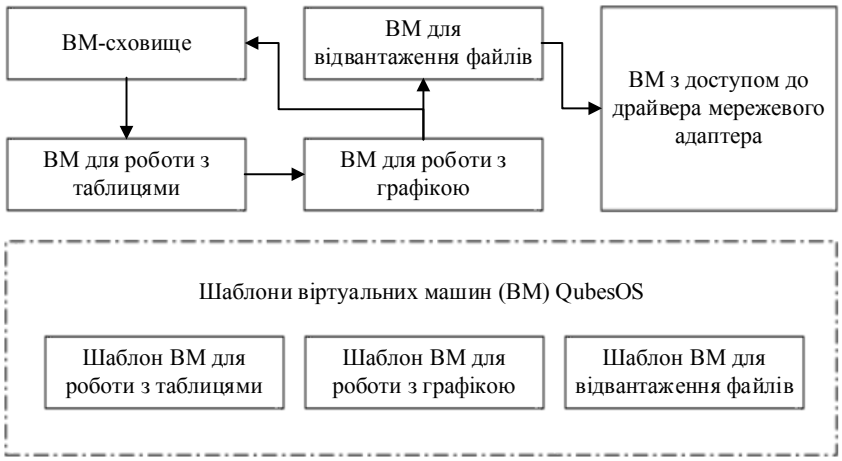

Рис.6. Схема сценарію роботи користувача із конфіденційними даними в умовах необхідності їх обробки

В результаті, кожне середовище має доступ лише до тих даних, з якими йому необхідно мати справу в конкретний момент часу за конкретних умов для виконання своїх цілей, а результати обробки безпечно зберігаються у спеціалізованому сховищі.

У рамках третього прикладу сценарію використання системи безпеки, користувачеві необхідно вести діяльність декількох типів у мережі Інтернет, не ідентифікуючи себе як одна і та ж особа. Для цього необхідно з технічної сторони, як мінімум, мати різні мережеві адреси свого пристрою. Досягти цього можна, як правило, шляхом використання проксі-сервісів. Але при традиційному підході постає така проблема, що скомпрометована система може бути уражена шкідливим ПЗ, яке матиме змогу контактувати с мережею Інтернет в обхід проксі-сервісів, що дасть змогу цьому ПЗ виявити реальні дані користувача, такі як MAC-адреса, IP-адреса, місце розташування та інші, чого як раз користувач намагався уникнути.

Для вирішення цієї проблеми користувачеві системи слід використовувати сценарій такого типу, як запропоновано на рис. 7, а саме: створювати ланцюг проксі-серверів на ізольованих машинах для отримання доступу у глобальну мережу на конкретних робочих машинах саме через ці ланцюги. Різни із традиційним підходом у тому, що навіть якщо робоча машина буде уражена шкідливим ПЗ, котре намагатиметься дізнатись приховані дані про пристрій користувача, цьому шкідливому ПЗ це не вдасться, оскільки такої інформації на робочій машині просто немає. Що і досягається за допомогою ізоляції проксі-ланцюга від робочої машини.

Більше того, спроектована система дозволяє ефективно використовувати саме ланцюг проксі-сервісів, а не лише один проксі сервіс. Практична вигода такого підходу полягає у тому, що кожен наступний проксі-сервіс зв'язується із попереднім, а не з робочою машиною, тому лише найближчий, у топології мереж, сервіс до віртуальної робочої машини знатиме якісь дані про неї. В результаті отримуємо, що навіть якщо скомпрометовано буде не робочу машину, а проксі-сервіс, то цей сервіс все-одно не отримає достатньої кількості даних щоб зрозуміти одночасно і ким являється клієнт (робоча машина), і до кого він звертається. Проксі-сервіс може знати тільки одне із цього, якщо знаходиться у топології біля робочої машини, або біля зовнішнього серверу, або не дізнається взагалі нічого із цього, якщо в топології знаходиться між двома іншими проксі-сервісами.

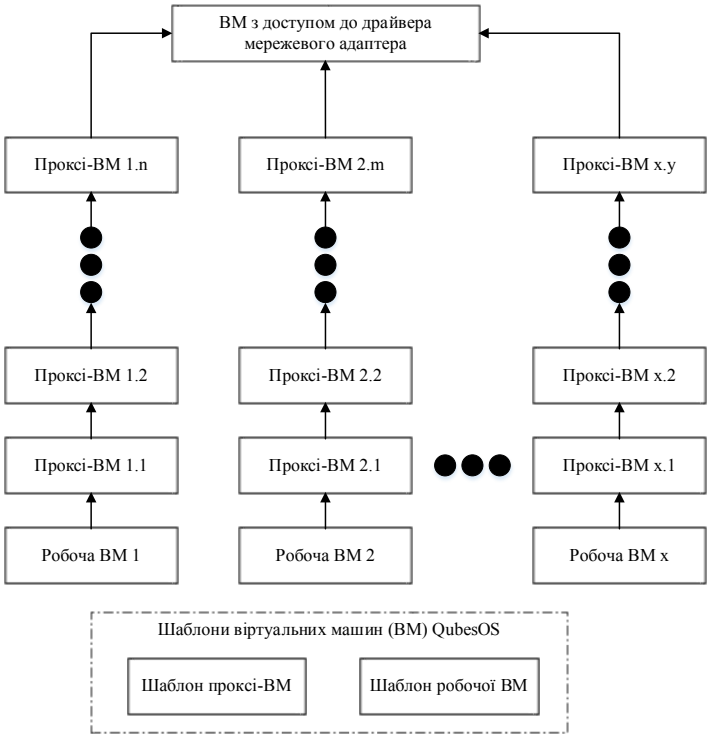

Рис. 7. Схема сценарію роботи користувача із кількома ізольованими діями у мережі Інтернет

У табл. 1. наведено порівняльний аналіз переваг і недоліків деяких типів систем із традиційним підходом до забезпечення безпеки, і пропонованої системи, на основі гібридного гіпервізора Xen, хостової операційної системи QubesOS, гостьових операційних систем у виді дистрибутивів GNU/Linux та додаткового програмного забезпечення.

Таблиця 1

Характеристики систем безпеки

Можливості та особливості систем	Тип системи							
	Win	Linux	ГРИФ	ЛОЗА	Win + ПЗ	Linux + ПЗ	X + Q	X + Q + ПЗ
Відкритий програмний код	Ні	Так	Ні	Ні	Ні	Так	Так	Так
Повністю безкоштовне користування	Ні	Так	Ні	Ні	Ні	Так	Так	Так
Надійне шифрування дисків	Ні	Ні	Ні	Ні	Так	Так	Так	Так
Ідентифікація користувача через пароль	Так	Так	Так	Так	Так	Так	Так	Так
Апаратний ключ доступу	Ні	Част.	Так	Так	Так	Так	Ні	Так
Надійне шифрування окремих файлів	Ні	Так	Ні	Ні	Так	Так	Так	Так
Обмеження на запуск конкретних програм	Ні	Так	Так	Так	Так	Так	Так	Так
Повне логування дій та процесів	Част.	Так	Част.	Част.	Част.	Так	Част.	Так
Ідентифікація периферії за незмінними параметрами	Ні	Част.	Так	Так	Так	Так	Ні	Так
Маркування друкуючихся документів	Ні	Так	Так	Ні	Так	Так	Ні	Так
Контроль експорту інформації на зйомні пристрої	Ні	Част.	Так	Так	Так	Так	Ні	Так
Контроль імпорту інформації зі зйомних пристроїв	Ні	Част.	Част.	Част.	Част.	Так	Ні	Так
Надійне безповоротнє затирання видалених файлів	Ні	Так	Так	Ні	Так	Так	Част	Так
Контроль доступу прикладних програм до файлів	Ні	Так	Так	Так	Так	Так	Так	Так
Контроль цілісності комплексу і окремих програм	Ні	Так	Так	Так	Част.	Так	Так	Так
Блокування системи на час відсутності користувача	Так	Так	Так	Так	Так	Так	Так	Так
Контроль цілісності системи і автодіагностика при старті	Ні	Част.	Част.	Част.	Част.	Так	Част.	Так
Створення резервних копій та відновлення по ним	Част.	Част.	Так	Так	Так	Так	Так	Так
Підтримка кількох користувачів	Так	Так	Так	Так	Так	Так	Ні	Ні
Віртуалізація із високою швидкодією	Ні	Ні	Ні	Ні	Ні	Ні	Так	Так
Ізоляція апаратних драйверів, розмежування доступу до них	Ні	Ні	Ні	Ні	Ні	Ні	Так	Так
Ізоляція периферійних пристроїв від складових системи	Ні	Ні	Ні	Ні	Ні	Ні	Так	Так
Підтримка довільної кількості одночасно працюючих ОС	Ні	Ні	Ні	Ні	Част.	Част.	Так	Так
Ізоляція мережевих налаштувань, брандмауера, політик безпеки	Ні	Ні	Ні	Ні	Ні	Ні	Так	Так
Безпечний ізольований запуск довільних програм	Ні	Ні	Ні	Ні	Част.	Част.	Так	Так
Безпечна ізольована робота з довільною ІзОД	Ні	Ні	Ні	Ні	Ні	Ні	Так	Так
Розмежування діяльностей на довільну кількість рівнів безпеки	Ні	Ні	Ні	Ні	Част.	Част.	Так	Так
Зручний уніфікований інтерфейс віртуалізації	Ні	Ні	Ні	Ні	Ні	Ні	Так	Так
Модифікація та інспекція будь-яких складових системи	Ні	Так	Ні	Ні	Ні	Так	Так	Так
Автоматизація будь-яких процесів, дій, налаштувань	Ні	Так	Ні	Ні	Част.	Так	Так	Так
Дистанційне адміністрування системи після встановлення	Ні	Част.	Так	Так	Так	Так	Ні	clip2net.com

В таблиці присутні наступні системи:

1. Win – система безпеки на базі ОС Windows будь-якої із версій без додаткового стороннього ПЗ;
2. Linux – система безпеки на базі таких дистрибутивів GNU/Linux, як Debian, Fedora, Archlinux, Whonix будь-якої із ПЗ, встановленим із офіційних стандартних репозиторіїв відповідного дистрибутиву, але без додаткового стороннього ПЗ, якого в тих репозиторіях немає;
3. ГРИФ – система безпеки на базі ОС Windows і встановленого на неї програмного комплексу захисту «ГРИФ» будь-якої із версій і варіантів;
4. ЛОЗА – система безпеки на базі ОС Windows і встановленого на неї програмного комплексу захисту «ЛОЗА» будь-якої із версій і варіантів;
5. Win + ПЗ – система захисту на базі ОС Windows та будь-якого додаткового стороннього ПЗ, встановленого на комп’ютер;
6. Linux + ПЗ – система захисту на базі вищезгаданих дистрибутивів GNU/Linux із будь-яким додатковим ПЗ;
7. X + Q – система на базі гіпервізора Xen, хостової ОС QuebesOS;

X + Q + ПЗ – система на базі гіпервізора Xen, хостової ОС QuebesOS, вищезгаданих дистрибутивів та будь-якого додаткового ПЗ, встановленого на комп’ютері.

ВИСНОВКИ

В роботі було проаналізовано недоліки традиційних підходів до забезпечення безпеки користувача ЕОМ від НСД до інформації, та з’ясовано, що вони не забезпечують достатній для підвищеної інформаційної безпеки рівень захисту.

Було проаналізовано існуючі типи і методи віртуалізації, обрано найбільш придатні для розробки системи захисту тип і метод. Було обрано придатний для реалізації обраного підходу до віртуалізації гіпервізор і хостову операційну систему, на базі якої доцільно базувати систему ізольованих віртуальних машин.

Було розроблено і реалізовано систему захисту на базі обраного підходу, а також протестовано на стійкість до певних видів атак, в результаті чого з’ясовано, що реалізована система є достатньо стійкою і забезпечує більший рівень безпеки, ніж аналоги.

Розроблена система захисту користувача від несанкіонованого доступу до інформації за допомогою ізоляції його дій у ізольованих паравіртуалізованих доменах засобами гібридного гіпервізора дає змогу:

1) ізолювати апаратні драйвери і мережеві ідентифікатори від конкретних віртуальних машин і загроз на них;

2) безпечно і зручно виконувати завідомо потенційно небезпечні дії на комп'ютері у одноразових віртуальних машинах;

3) одночасно безпечно використовувати велику кількість різних операційних систем кількох сімейств на одному комп'ютері із зручним інтерфейсом і високою швидкодією;

4) розмежовувати дії користувача на домени із різними рівнями безпеки.

СПИСОК ЛІТЕРАТУРИ

1. Stallings W. Computer security : principles and practice 3d edition / W. Stallings, L. Brown., 2015. – 842 с.
2. Aycock J. Computer Viruses and Malware / John Aycock., 2010. – 228 с.
3. Szor P. The Art of Computer Virus Research and Defense / Peter Szor., 2005.
4. Antivirus [Електронний ресурс] // The Virus Encyclopedia. – 2014. – Режим доступу до ресурсу: http://virus.wikidot.com/antivirus.
5. Выбор антивирусной защиты [Електронний ресурс] // AO Kaspersky Lab. – 2016. – Режим доступу до ресурсу: https://securelist.ru/threats/vybor-antivirusnoj-zashhity/.
6. Качество антивирусной защиты и проблемы антивирусных программ [Електронний ресурс] // AO Kaspersky Lab. – 2017. – Режим доступу до ресурсу: https://securelist.ru/threats/kachestvo-antivirusnoj-zashhity-i-problemy-antivirusnyx-programm/.
7. Варганов Д. С. классификация средств защиты программного обеспечения [Текст] / Д. С. Варганов. – Москва, 2007. – 153 с.
8. An Introduction to Qubes OS [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://www.qubes-os.org/intro/#arent-antivirus-programs-and-firewalls-enough.

REFERENCES

1. Stallings W. Computer security: principles and practice 3d edition / W. Stallings, L. Brown, 2015. – 842 s .
2. Aycock J. Computer Viruses and Malware / John Aycock., 2010. – 228 p.
3. Szor P. The Art of Computer Virus Research and Defense / Peter Szor., 2005.
4. Antivirus [Electronic resource] // The Virus Encyclopedia. – 2014. – Resource access mode: http://virus.wikidot.com/antivirus.
5. Selection of antivirus protection [Electronic resource] // AO Kaspersky Lab. – 2016 – Resource access mode: https://securelist.ru/threats/vybor-antivirusnoj-zashhity/.
6. The quality of antivirus protection and the problem of antivirus programs [Electronic resource] // AO Kaspersky Lab. – 2017. – Resource access mode: https://securelist.ru/threats/kachestvo-antivirusnoj-zashhity-i-problemy-antivirusnyx-programm/.
7. Varganov D. S. classification of software protection means [Text] / D. S. Varganov. – Moscow, 2007. – 153 s .
8. An Introduction to Qubes OS [Electronic resource]. – 2017. – Resource access mode: https://www.qubes-os.org/intro/#arent-antivirus-programs-and-firewalls-enough.

Надійшла до редакції 03.12.2017р.

КАРПІНЕЦЬ ВАСИЛЬ ВАСИЛЬОВИЧ – к.т.н., доцент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет.

КОСТЮЧЕНКО ОЛЬГА ІГОРІВНА – магістрант кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет.

ПАВЛОВСЬКИЙ ПАВЛО ВАЛЕРІЙОВИЧ – асистент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет.

ПРИЙМАК АНДРІЙ ВАСИЛЬОВИЧ – аспірант кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет.

ЮХИМЕНКО СВЯТОСЛАВ ВАЛЕНТИНОВИЧ – магістрант кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет.